JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 06282440
(43)Date of publication of application: 07.10.1994

(51)Int.Cl. G06F 9/45

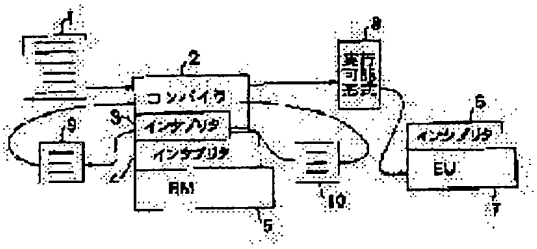
(21)Application number: 05066947
(22)Date of filing: 25.03.1993

(71)Applicant: SONY CORP
(72)Inventor: KITATANI YOSHIMICHI

(54) PROGRAM EDITING METHOD, PROGRAM EDITING AND EXECUTING METHOD, AND COMPUTER SYSTEM

(57)Abstract:

PURPOSE: To provide a processing system which edits a program and the program editing method which can efficiently edit machine codes that can be executed on a processing system different from the processing system.
CONSTITUTION: A program processing system RM which has a compiler 2 for editing the source program I and generates the machine codes and a program processing system EU which executes the generated machine codes are provided. The processing system RM is provided with an interpreter 4 and an interpreter 3 and a compiler 2 which operate on the interpreter 4. The interpreter 3 simulates the operation of an interpreter 6 which operates on the processing system EU. The compiler 2 reads in the source program 1 of a LISP language process and generates a program 10 to be run on the interrupter 3. The interpreter 3 executes the program 10 instead of the interpreter 6 and offers program information which can not be execute to the compiler 2 as a program 9. The compiler 2 further edits the program 9. The interpreter 4 executes the generated machine codes 8 on the processing system EU.



LEGAL STATUS

[Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（Ａ）**　(11)特許出願公開番号

特開平6－282440

(43)公開日　平成６年(1994)10月７日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　9/45				
		9292－5B	Ｇ０６Ｆ　9/44　３２０　Ａ	

審査請求　未請求　請求項の数８　ＯＬ　（全　９　頁）

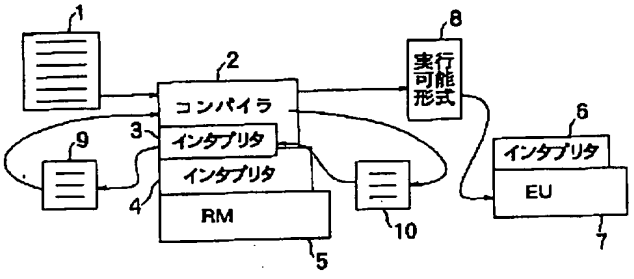
(21)出願番号　特願平5－66947

(22)出願日　平成５年(1993)３月25日

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川６丁目７番35号
(72)発明者　北谷　義道
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(74)代理人　弁理士　佐藤　隆久

(54)【発明の名称】　プログラム編集方法、プログラム編集・実行方法およびコンピュータシステム

(57)【要約】
【目的】　プログラム編集する処理系と、この処理系とは異なる処理系で実行可能な機械語コードを効率よく編集可能なプログラム編集方法を提供する。
【構成】　ソースプログラム１１を編集して機械語コードを生成するコンパイラ１を有するプログラム処理系ＲＭと、生成された機械語コードを実行するプログラム処理系ＥＵと有する。処理系ＲＭには、インタプリタ４、このインタプリタ４の上で動作するインタプリタ３およびコンパイラ２が設けられている。インタプリタ３は処理系ＥＵで動作するインタプリタ６の動作を模擬する。コンパイラ２はＬＩＳＰ言語処理のソースプログラム１を読み込んで、インタプリタ３で動作させるプログラム１０を生成する。インタプリタ３はインタプリタ６に代わってプログラム１０を実行して、実行できないプログラム情報をプログラム９としてコンパイラ２に提供する。コンパイラ２はそのプログラム９をさらに編集する。インタプリタ４は生成された機械語コード８を処理系ＥＵにおいて実行する。



【特許請求の範囲】
【請求項１】プログラム実行の前に、実行されるプログラム自身を編集し変換する前処理を、自身で記述できる言語要素を備えたプログラミング言語で記述されたソースプログラムについて、
そのソースプログラムが実行されるプログラム処理系と別のプログラム処理系で動作し、
プログラムが実行されるプログラム処理系の機能を擬似的に実現する処理機能を有し、
前記擬似実行によって前処理を行なうことを特徴とするプログラム編集方法。
【請求項２】機械語コードに変換される際に、実行されるプログラム自身を編集し変換する前処理を、自身で記述できる言語要素を備えたプログラミング言語で記述されたソースプログラムについて、
擬似実行による前処理を行ないながら機械語コードに変換することを特徴とする請求項１記載のプログラム編集方法。
【請求項３】ソースプログラムの実行とは異なったプログラム処理系で前処理が実行され、
プログラムが実行されるプログラム処理系に対応した次段階のプログラム言語で記述されたプログラムを作成することを特徴とする請求項１記載のプログラム編集方法。
【請求項４】前記ソースプログラムがＬＩＳＰ言語で生成されている、請求項１～３いずれか記載のプログラム編集方法。
【請求項５】第１のプログラム処理系で動作するプログラム編集処理プログラムと、第１のインタプリタと第２のインタプリタを有し、
第１のプログラム処理系に接続された第２のプログラム処理系で動作する第３のインタプリタを有し、
前記プログラム編集処理プログラムは、プログラム実行の前に、実行されるプログラム自身を編集し変換する前処理を、自身で記述できる言語要素を備えたプログラミング言語で記述されたソースプログラムに編集し、第１のインタプリタ言語および第２のプログラム処理系で実行される機械語コードに変換する機能を有し、
第２のインタプリタは、前記プログラム編集プログラムが編集されて生成された第１のインタプリタ言語を、第２のプログラム処理系における第３のインタプリタの動作を模擬して実行して第２のインタプリタ言語を生成し、
第３のインタプリタは該第１のインタプリタ言語を処理してプログラム編集情報を前記プログラム編集処理プログラムに提供し、
前記プログラム編集処理プログラムが該プログラム編集情報を編集して、第２のプログラム処理系における第３のインタプリタで実行される前記機械語コードを生成し、
第３のインタプリタは該機械語コードを前記第２のプログラム処理系において実行するプログラム編集・実行方法。
【請求項６】前記ソースプログラムがＬＩＳＰ言語で生成されている、請求項５記載のプログラム編集・実行方法。
【請求項７】第１のプログラム処理系と該第１のプログラム処理系に接続された第２のプログラム処理系とを有し、
第１のプログラム処理系は、プログラム編集処理プログラムと、第１のインタプリタと第２のインタプリタを有し、
第２のプログラム処理系は第３のインタプリタを有し、
前記プログラム編集処理プログラムは、プログラム実行の前に、実行されるプログラム自身を編集し変換する前処理を、自身で記述できる言語要素を備えたプログラミング言語で記述されたソースプログラムに編集し、第１のインタプリタ言語および第２のプログラム処理系で実行される機械語コードに変換する機能を有し、
第２のインタプリタは、前記プログラム編集プログラムが編集をされて生成された第１のインタプリタ言語を、第２のプログラム処理系における第３のインタプリタを模擬して実行して第２のインタプリタ言語を生成し、
第３のインタプリタは第２のインタプリタ言語を処理してプログラム編集情報を前記プログラム編集処理プログラムに提供し、
前記プログラム編集処理プログラムが該プログラム編集情報を編集して、前記機械語コードを生成し、
第３のインタプリタは該機械語コードを前記第２のプログラム処理系において実行するコンピュータシステム。
【請求項８】前記ソースプログラムがＬＩＳＰ言語で生成されている、請求項７記載のコンピュータシステム。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】本発明は、デジタル計算機におけるプログラムを編集する方法に関するものであり、特に、実行させる処理系とは異なる処理系で編集を行なうプログラム編集方法、プログラム編集・実行方法、および、これらを行うコンピュータシステムに関する。
【０００２】
【従来の技術】従来から、Ｃ言語などの高級プログラミング言語で記述されたソースプログラムを機械語に変換するといったコンパイラと、生成された機械語を読み取り実行する処理系とは別の処理系で動作させるようなコンパイラであるクロスコンパイラとを用いるプログラム編集方法が知られており、実際に使用されている。以下、クロスコンパイラを使用し、高級プログラム言語で記述されたソースプログラムを下位言語に変換する、従来のプログラム編集方法について述べる。
【０００３】図５は従来のクロスコンパイラのデータの

流れを示した図である。図５においてソースプログラム１１はＣ言語で記述されたプログラムである。コンパイラ１２はプログラム処理系（Ａ）１３上で動作し、ソースプログラム１１を読み込んでコンピュータで実行可能な機械語コード１４を生成する。この機械語コード１４はプログラム処理系（Ｂ）１５に読み込まれて実行される。

【０００４】また、クロスコンパイルと呼ばれるプログラム編集方法とは別に、従来から、図６に示したように、１つの処理系の中にソースプログラム１１を読み込み、機械語コード１４に変換しながら随時プログラムの実行を行なうインタプリタ１６と、ソースプログラム１１から機械語コード１４に一括してプログラムを編集するコンパイラ１２が１つの処理系内に同時に存在し、コンパイル時にインタプリタ１６にコンパイル処理の一部として実行されるプログラムの変換処理を記述できるプログラミング言語と、ソースプログラム中の指定された処理をコンパイル中にインタプリタ１６に実行させるようなプログラム編集方法が知られており、実際に使用されている。

【０００５】図６においてソースプログラム１１はＬＩＳＰで記述されたプログラムである。コンパイラ１２はＬＩＳＰインタプリタ１６上で動作し、ソースプログラム１１を読み込んで実行可能な機械語コード１４を生成する。コンパイラ１２はソースプログラム１１中にコンパイル時の実行を指定するプログラムを見つけると、プログラム１７としてインタプリタ１６に渡し実行させる。インタプリタ１６はその結果生じる出力を新たなプログラム１８としてコンパイラ１２に再び戻す。コンパイラ１２はプログラム１８を受け取るとそれを機械語コード１４に変換する。

【０００６】

【発明が解決しようとする課題】しかしながら、図６を参照して上述した従来のプログラム編集方法では、一つの処理系の中にインタプリタ１６とコンパイラ１２が同時に存在し、コンパイル時にインタプリタ１６にコンパイル処理の一部として実行させる処理を記述できるプログラミング言語で書かれたソースプログラム１１を、そのソースプログラム１１から生成された機械語コード１４が実行されるプログラム処理系（図示せず）とは別の処理系、この例ではフレーム処理系（Ａ）１３で、機械語コード５に変換することはできない。そのため、従来のプログラム編集方法では、プログラム処理系を開発する場合、プログラム処理系がコンパイラを動作させるだけ充分な機能を持つ前にプログラム処理系の機能をコンパイラによって作成し、効率的に開発を行なうことが出来ないという問題が生じる。

【０００７】また、従来のプログラム編集方法では、より高速な処理系上のコンパイラを用いて機械語コードを作成し、より効率的にプログラム開発を行なうことができないという問題が生じる。

【０００８】したがって、本発明は、プログラムを編集するプログラム処理系とは異なるプログラム処理系で動作可能な機械語コードを生成可能なプログラム編集方法を提供することを目的とする。また本発明は、効率良くプログラム開発可能なフレーム編集方法を提供することを目的とする。

【０００９】さらに本発明は、上述したプログラム編集およびその実行が可能なコンピュータシステムを提供することを目的とする。

【００１０】

【課題を解決するための手段、および、作用】上述した問題を解決するため、本発明のプログラム編集方法においては、プログラム実行の前に、実行されるプログラム自身を編集し変換する前処理を、自身で記述できる言語要素を備えたプログラミング言語で記述されたソースプログラムについて、そのソースプログラムが実行されるプログラム処理系と別のプログラム処理系で動作し、プログラムが実行されるプログラム処理系の機能を擬似的に実現する機能を有し、擬似実行によって前処理を行なうようにした。

【００１１】また、本発明のプログラム編集方法においては、機械語コードに変換される際に、プログラム自身を編集し変換する前処理を、自身で記述できる言語要素を備えたプログラミング言語で記述されたソースプログラムについて、疑似実行による前処理を行ないながら機械語コードに変換するようにした。

【００１２】さらに、本発明のプログラム編集方法においては、ソースプログラムの実行とは異なったプログラム処理系（計算機）で動作し、プログラムが実行される処理系の機能を疑似的に実現する機能を有し、疑似実行によって前処理を行ない、プログラムが実行されるプログラム処理系（計算機）に対応した次段階のプログラム言語で記述されたプログラムを作成するようにした。

【００１３】上記ソースプログラムとしては、好適には、ＬＩＳＰ言語で記載されたプログラムである。

【００１４】また本発明によれば、第１のプログラム処理系で動作するプログラム編集処理プログラムと、第１のインタプリタと第２のインタプリタを有し、第１のプログラム処理系に接続された第２のプログラム処理系で動作する第３のインタプリタを有し、前記プログラム編集処理プログラムは、プログラム実行の前に、実行されるプログラム自身を編集し変換する前処理を、自身で記述できる言語要素を備えたプログラミング言語で記述されたソースプログラムに編集し、第１のインタプリタ言語および第２のプログラム処理系で実行される機械語コードに変換する機能を有し、第２のインタプリタは、前記プログラム編集プログラムが編集されて生成された第１のインタプリタ言語を、第２のプログラム処理系における第３のインタプリタの動作を模擬して実行して第２

のインタプリタ言語を生成し、第３のインタプリタは該第１のインタプリタ言語を処理してプログラム編集情報を前記プログラム編集処理プログラムに提供し、前記プログラム編集処理プログラムが該プログラム編集情報を編集して、第２のプログラム処理系における第３のインタプリタで実行される前記機械語コードを生成し、第３のインタプリタは該機械語コードを前記第２のプログラム処理系において実行するプログラム編集・実行方法が提供される。

【００１５】さらに、本発明によれば、第１のプログラム処理系と該第１のプログラム処理系に接続された第２のプログラム処理系とを有し、第１のプログラム処理系は、プログラム編集処理プログラムと、第１のインタプリタと第２のインタプリタを有し、第２のプログラム処理系は第３のインタプリタを有し、前記プログラム編集処理プログラムは、プログラム実行の前に、実行されるプログラム自身を編集し変換する前処理を、自身で記述できる言語要素を備えたプログラミング言語で記述されたソースプログラムに編集し、第１のインタプリタ言語および第２のプログラム処理系で実行される機械語コードに変換する機能を有し、第２のインタプリタは、前記プログラム編集プログラムが編集をされて生成された第１のインタプリタ言語を、第２のプログラム処理系における第３のインタプリタを模擬して実行して第２のインタプリタ言語を生成し、第３のインタプリタは第２のインタプリタ言語を処理してプログラム編集情報を前記プログラム編集処理プログラムに提供し、前記プログラム編集処理プログラムが該プログラム編集情報を編集して、前記機械語コードを生成し、第３のインタプリタは該機械語コードを前記第２のプログラム処理系において実行するコンピュータシステムが提供される。

【００１６】

【実施例】図面を参照して本発明の実施例について説明する。図１は、本発明のプログラム編集方法が適用されるプロセッサエレメントＰＥ（コンピュータシステム）の概略構成図である。図１（Ａ）は単独のプロセッサエレメントＰＥの構成を示す図であり、図１（Ｂ）は複数のプロセッサエレメントＰＥが相互に接続されて大規模コンピュータシステムを構成する図を示す。

【００１７】図１（Ａ）に示したプロセッサエレメントＰＥは、ＬＩＳＰなどの記号言語を処理する記号処理用計算機エバリュエータＥＵと入出力や記憶管理に使われる計算機ＲＭと主記憶ＭＭおよび二次記憶装置とを有するストーリジ（記憶装置）から構成されている。記号処理用計算機エバリュエータＥＵに対して、入出力や記憶管理に使われる計算機ＲＭおよびストーリジが一種のリソース（資源）として扱われている。

【００１８】図１（Ｂ）は、各プロセッサエレメントＰＥ内の入出力や記憶管理に使われる計算機ＲＭが、ＬＡＮ（Local Area Network）などの相互接続系（インターリソースコネクション）を介して、相互に接続されて、複数のプロセッサエレメントＰＥが全体として、大規模コンピュータシステムを構成している。

【００１９】図２は図１（Ａ）に示されたプロセッサエレメント（ＰＥ）の構成の詳細を示す図である。本発明のプログラム編集方法は、図２に示すような記号処理用計算機エバリュエータＥＵと入出力や記憶管理に使われる計算機（ＲＭ）からなる並列計算機ネットワークのプロセッサエレメントＰＥに適用される。図５および図６に対応させると、入出力や記憶管理に使われる計算機ＲＭがプログラム処理系Ａに対応し、入出力や記憶管理に使われる計算機ＲＭがプログラム処理系Ｂに対応している。

【００２０】記号処理用計算機エバリュエータＥＵは、データ幅３２ビットのＶＬＩＷ（Very Long Instruction Word）制御のプロセッサであり、ユーザ定義の関数やメソッドのコンパイルされたコードを高速実行するためのインストラクションキャッシュ（６４ビット×６４Ｋワード）を持つ。また、記号処理用計算機エバリュエータＥＵは、データのフィールド演算を高速に実現するためのマスカ（Ｍａｓｋｅｒ）／算術論理処理ユニット（ＡＬＵ）を持ち、関数やメソッドの呼び出しを高速化するためのデータフィールド値による多方向分岐機能を持ち、さらに多方向分岐のためのルックアップテーブルと記号処理用計算機エバリュエータＥＵのシステムスタックのための６４Ｋワードの制御用メモリを持つ。さらに記号処理用計算機エバリュエータＥＵ、記号処理計算を実行する際のフレームスタックとレジスタファイルのための６４Ｋワードのローカルメモリを持つ。記号処理用計算機エバリュエータＥＵは、入出力や記憶管理に要する演算は、入出力や記憶管理に使われる計算機ＲＭにその実行を要求する。

【００２１】入出力や記憶管理に使われる計算機ＲＭは、記憶管理、デバイス管理、プロセス管理、ファイル管理、入出力、他のプロセッサエレメントＰＥとの通信の機能を記号処理用計算機エバリュエータＥＵに提供する。入出力や記憶管理に使われる計算機ＲＭには、ワークステーションＷＳを使用して、ワークステーションＷＳのＸウインドウ・システム上にアプリケーション開発時および実行時のインタラクティブなヒューマンインタフェース環境を実現する。そのため、入出力や記憶管理に使われる計算機ＲＭは関数定義やメソッド定義のインクリメンタルな修正・コンパイル、エラー状態からの復帰・実行再開機能を持つ。さらに入出力や記憶管理に使われる計算機ＲＭは、アプリケーションの実行時にユーザとの動的な対話ができるような入出力機能を提供する。入出力や記憶管理に使われる計算機ＲＭは、主記憶ＭＭ上の記号処理データの高速アクセスのためのポイント・マニュプレータ（Pointer Manipulator）を持つ。入出力や記憶管理に使われる計算機ＲＭは、並列処理時

に他のプロセッサエレメントＰＥとの通信を行なうための通信インタフェースを装着可能である。

【００２２】記憶装置（ストーリジ）のうち主記憶MMは、記号処理用計算機エバリュエータＥＵからも直接アクセス可能なデュアルポートメモリである。主記憶MMでは、排他的アクセス、記号処理演算で有効な参照カウントなどの機能アクセスを提供する。主記憶MMは、これらの制御情報と２ワードのポインタを表現するため、８０ビット×２５６Ｋワードの構成となっている。

【００２３】記憶装置のうち二次記憶は、ワークステーションＷＳのディスクシステムを使用する。

【００２４】記号処理用計算機エバリュエータＥＵと入出力や記憶管理に使われる計算機ＲＭとは、入出力や記憶管理に使われる計算機インターフェースＲＭ－Ｉ／Ｆで接続され、記号処理用計算機エバリュエータＥＵから入出力や記憶管理に使われる計算機ＲＭに、また入出力や記憶管理に使われる計算機ＲＭから記号処理用計算機エバリュエータＥＵに機能データを送出する。また、記号処理用計算機エバリュエータＥＵからメインメモリインターフェースMM－Ｉ／Ｆを介して情報データが送出され、主記憶MMと入出力や記憶管理に使われる計算機ＲＭに送出される。

【００２５】入出力や記憶管理に使われる計算機インタフェースＲＭ－Ｉ／Ｆ、および、主記憶インタフェースMM－Ｉ／Ｆは、記号処理用計算機エバリュエータＥＵ内の内部バスＩＢＵＳを介して算術論理処理ユニットＡＬＵに接続される。また、入出力や記憶管理に使われる計算機インタフェースＲＭ－Ｉ／Ｆは、入出力や記憶管理に使われる計算機ＲＭ内のエバリュエータインターフェースＥＵ－Ｉ／Ｆを介してワークステーションＷＳに接続される。

【００２６】図３は本発明の実施例のコンパイラのソフトウェア構成を表す図である。入出力や記憶管理に使われる計算機（ＲＭ）５が、図５および図６におけるプログラム処理系Ａに対応し、記号処理用計算機エバリュエータ（ＥＵ）７がプログラム処理系Ｂに対応している。入出力や記憶管理に使われる計算機（ＲＭ）５には、コンパイラ２、第１のインタプリタ３および第２のインタプリタ４が入出力や記憶管理に使われる計算機（ＲＭ）５の上で動作し、実行可能な機械コード８を生成する。記号処理用計算機エバリュエータ（ＥＵ）７が実行可能な機械語コード８を実行させる第３のインタプリタ６を内蔵している。

【００２７】ソースプログラム１は、ＬＩＳＰで記述されたソースプログラムである。インタプリタ４は入出力や記憶管理に使われる計算機ＲＭ５上で動作しするインタプリタで、コンパイラ２およびインタプリタ３はこのインタプリンタ４上で動作する。入出力や記憶管理に使われる計算機（ＲＭ）５は、図２に示した計算機（ＲＭ）である。インタプリタ３は、記号処理用計算機エバリュエータＥＵ上のインタプリタ６の動作を疑似的に実行するインタプリタであり、コンパイラ２が処理中に呼び出でされる。コンパイラ２は、インタプリタ３、インタプリタ４とともにＬＩＳＰで記述されたソースプログラム１の解析を行ない、記号処理用計算機エバリュエータ（ＥＵ）７において実行可能な形式の機械語コード８を生成する。機械語コード８は記号処理用計算機エバリュエータ（ＥＵ）７に読み込まれ実行される。記号処理用計算機エバリュエータＥＵは、図２のエバリュエータＥＵである。

【００２８】図４は本発明のコンパイラのプログラム編集方法およびその実行を示すフローチャートである。入出力や記憶管理に使われる計算機（ＲＭ）５上のコンパイラ２は、ステップ０１（Ｓ０１）において起動されると、ステップ０２（Ｓ０２）においてソースプログラム１からプログラムの一部を取り込み、プログラムがある限り以下の処理、ステップ０４～０５、０６～０７を繰り返す。プログラム処理が終了すると、ステップ０３（Ｓ０３）において、プログラム編集処理が終了する。

【００２９】Ｓ０４において取り込んだ入力がインタプリタ３に前処理の実行を指定した記述である場合、コンパイラ２は、そのプログラムの部分１０を第１のインタプリタ言語として、記号処理用計算機エバリュエータＥＵにおけるインタプリタ６の模擬動作を行うインタプリタ３に渡す。Ｓ０４において取り込んだ入力がインタプリタ３に前処理の実行を指定した記述でない場合、コンパイラ２は、入力されたプログラム１に対して機械語コード８を生成する。インタプリタ３は、Ｓ０６においてそのプログラムを実行し、その結果として編集用プログラム９を生成する。コンパイラ２はＳ０７において編集用プログラム９を新たに入力として取り込み、その編集用プログラム９に対してさらにコンパイル処理を行なう。

【００３０】このように、本発明の実施例によれば、プログラムが実行される処理系である記号処理用計算機エバリュエータＥＵとは別の処理系である入出力や記憶管理に使われる計算機ＲＭでプログラミングを編集することを可能とし、プログラム処理系開発の初期段階における開発効率を向上させることが出来る。また、本発明のプログラム編集方法によれば、より高速な計算機上で目的プログラムの編集を行ない、開発の効率を向上させることが出来る。

【００３１】さらに本発明の実施例によれば、疑似実行による変換を行ないながら機械語に変換することにより、プログラムが実行される処理系とは別の処理系でプログラムを編集することを可能とし、処理系開発の初期段階における開発効率を向上させることが出来る。あるいは、より高速な計算機上で目的プログラムの編集を行ない、プログラム開発の効率を向上させることが出来る。

【００３２】上述した実施例は、図１（Ａ）に示した単独のプロセッサエレメントＰＥにおけるプロセッサエレメントＰＥ編集処理およびその実行を例示したが、本発明のプログラム編集方法は、図１（Ｂ）示したコンピュータシステムにおいても，適用できる。たとえば、第１のプロセッサエレメントＰＥ内の入出力や記憶管理に使われる計算機ＲＭをプログラム処理系Ａとし、この入出力や記憶管理に使われる計算機ＲＭにおいて上述したプログラム編集処理を行い、その結果としての機械コードをＬＡＮなどで接続された第２のプロセッサエレメントＰＥ内の記号処理用計算機エバリュエータＥＵまたは入出力や記憶管理に使われる計算機ＲＭにおいて実行させることもできる。

【００３３】

【発明の効果】本発明のプログラム編集方法によれば、プログラム実行の前処理を自身で記述できる言語要素を備えた高級プログラミング言語で記述されたソースプログラムについて、そのソースプログラムが実行される処理系と別の処理系で動作し、プログラムが実行される処理系の機能を疑似的に実現する機能を有し、疑似実行によってソースプログラムの前処理を行ってから機械語に変換することにより、プログラムが実行される処理系とは別の処理系でプログラミングを編集することを可能とし、処理系開発の初期段階における開発効率を向上させることが出来る。

【００３４】また、本発明のプログラム編集方法によれば、より高速な計算機上で目的プログラムの編集を行ない、開発の効率を向上させることが出来る。

【００３５】さらに本発明のプログラム編集方法によれば、機械語コードに変換される際に自身で記述された機能を実行する必要のある高級プログラミング言語で記述されたソースプログラムについて、そのソースプログラムが実行される処理系と別の処理系で動作し、プログラムが実行される処理系の機能を疑似的に実現する機能を有し、疑似実行による変換を行ないながら機械語コードに変換することにより、プログラムが実行される処理系とは別の処理系でプログラムを編集することを可能とし、処理系開発の初期段階における開発効率を向上させることが出来る。あるいは、より高速な計算機上で目的プログラムの編集を行ない、プログラム開発の効率を向上させることが出来る。

【００３６】本発明によれば、プログラム編集と実行を連続的に行うプログラム編集・実行方法が提供できる。

【００３７】さらに本発明によれば、上記プログラム編集・実行を行うコンピュータシステムを提供できる。

【図面の簡単な説明】

【図１】本発明が適用される計算機ネットワークの接続形態を示す図であり、（Ａ）は、プロセッサエレメントＰＥ単体の構成図であり、（Ｂ）は複数のプロセッサエレメントＰＥが大規模コンピュータシステムを構成する図である。

【図２】図１に示したプロセッサエレメントＰＥの詳細構成を示す図である。

【図３】本発明のコンパイラのソフトウェア構成を表す図である。

【図４】本発明のプログラム編集・実行をの処理のフローチャートである。

【図５】従来のクロスコンパイラのソフトウェア構成を示す図である。

【図６】従来のインタプリタとコンパイラのソフトウェア構成を示す図である。

【符号の説明】

１・・プログラム編集対象のプログラム

２・・コンパイラ

３・・記号処理用計算機エバリュエータのインタプリタの動作を模擬するインタプリタ

４・・入出力や記憶管理に使われる計算機ＲＭの上で動作する基本的なインタプリンタ

５・・入出力や記憶管理に使われる計算機ＲＭ

６・・記号処理用計算機エバリュエータＥＵの上で動作するインタプリンタ

７・・記号処理用計算機エバリュエータＥＵ

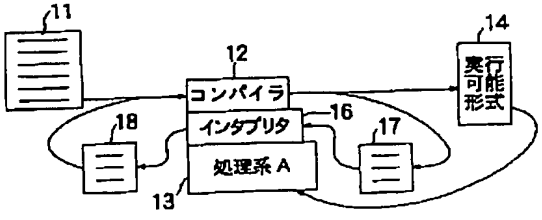
８・・記号処理用計算機エバリュエータＥＵで実行可能な機械語コード

９・・インタプリンタ４で生成された編集用プログラム

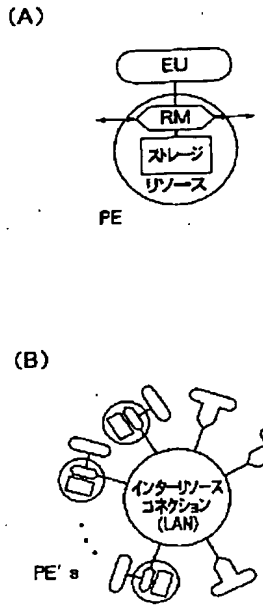
ＲＭ・・入出力や記憶管理に使われる計算機（プログラム処理系Ａ）

ＥＵ・・記号処理用計算機エバリュエータ（プログラム処理系Ｂ）

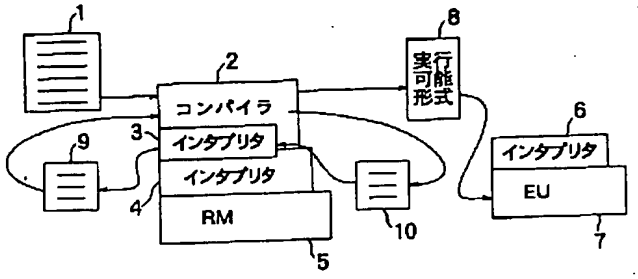
【図６】



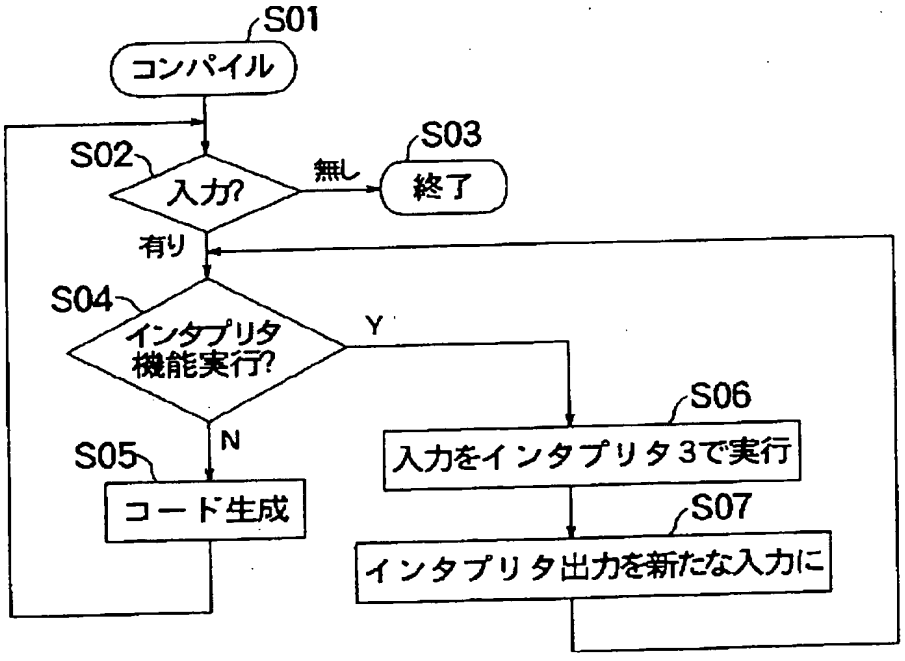
【図１】


(A)
EU
RM
ストレージ
リソース
PE
(B)
インターリソース
コネクション
(LAN)
PE's

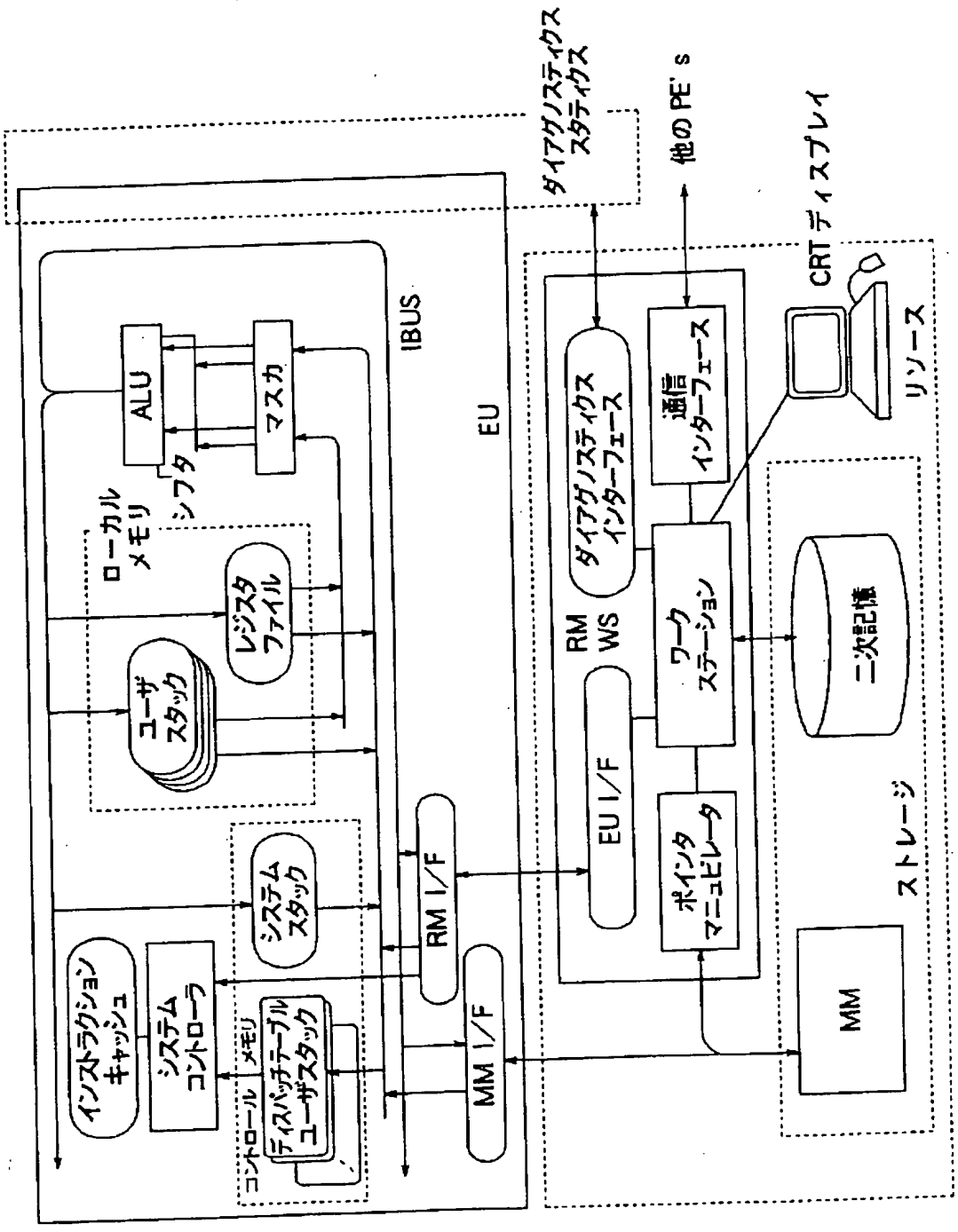
【図３】


1
2
コンパイラ
8
実行可能形式
6
インタプリタ
EU
7
3
インタプリタ
インタプリタ
9
4
RM
5
10

【図４】


S01
コンパイル
S02
入力?
無し
S03
終了
有り
S04
インタプリタ
機能実行?
Y
N
S05
コード生成
S06
入力をインタプリタ3で実行
S07
インタプリタ出力を新たな入力に

【図2】


ダイアグノスティクス
スタティクス
他のPE's
CRTディスプレイ
リソース
IBUS
EU
ALU
マスカ
シフタ
ローカル
メモリ
レジスタ
ファイル
ユーザ
スタック
システム
スタック
RM I/F
MM I/F
インストラクション
キャッシュ
システム
コントローラ
コントロール メモリ
ディスパッチテーブル
ユーザスタック
ダイアグノスティクス
インターフェース
通信
インターフェース
RM
WS
EU I/F
ワーク
ステーション
ポインタ
マニュピレータ
二次記憶
MM
ストレージ

【図5】